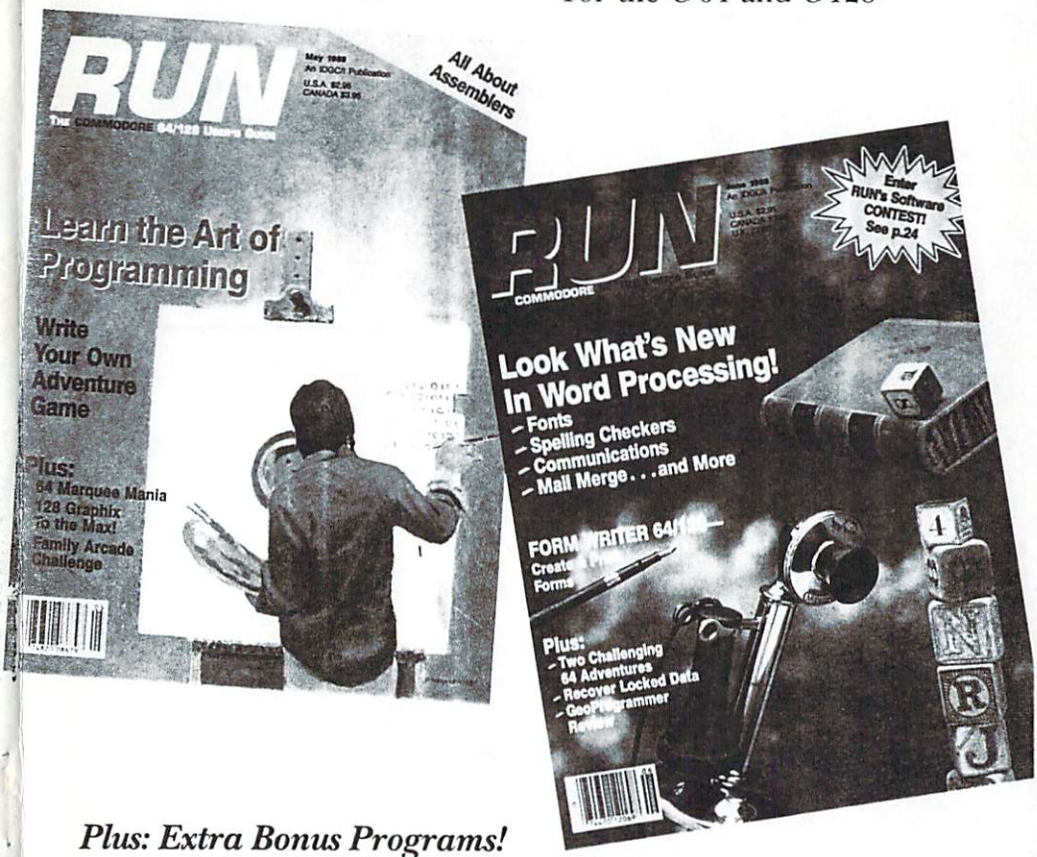
May/June 1988 Edition

RE RUN

RUN Programs on Disk

For the C-64 and C-128



Plus: Extra Bonus Programs!

Introduction

May-June '88 ReRUN

MOST OF THE MANY Commodore user group newsletters I have the fun of reading are exclusively devoted to the C-64 and C-128, but a few also offer some Amiga and MS-DOS coverage. I welcome this approach because it serves to remind readers that if you want good software at reasonable prices, the C-64 and C-128 are still your best bet. If you don't believe me, call your local computer store and ask them how much a copy of Lotus 1-2-3 costs for a PC clone. The store I called quoted me a retail price of $495.00!

I'm an advocate of good value for a fair price, and I feel that this philosophy is reflected in ReRUN, which always delivers plenty of varied, high-quality programs for your money. This issue is not an exception.

We lead off with one of my favorites, Moving Messages, for the C-64. Originally published in the May issue of *RUN*, Moving Messages lets you create huge screen characters that move across the screen. You'll find this program handy for classroom and showroom displays.

Another exciting program from the May issue is Drip-Drop. Here, you select a skill level between 1 (lowest) and 25, and then you attempt to catch falling raindrops with a pan before they hit a campfire at the bottom of the screen. Should a drop of water hit the fire, you lose one of your three pans.

An unusual program, also from the May issue, is Graphix to the Max, which allows you to display C-64 screen images on the C-128's 80-Column screen. You'll find that Doodle! pictures convert especially well with this program.

Tick-Tock 128, May's Easy Applications program, gives your 128 an extremely accurate analog clock in 40-Column mode. It comes complete with chimes that sound a lovely melody every quarter hour.

Knowledgable C-64 owners are aware of the potential dangers that are associated with employing the Save-With-Replace (SAVE"@0:FILENAME",8) command on the C-64. Sometimes multiple files are corrupted by using this command. May's Mega-Magic entry, Scratch and Save, eliminates these worries by creating an all-new Save-With-Replace command.

The roundup of June programs begins with a versatile program called Form Writer. With it, you can create letters and forms, read in lists of names and addresses from its built-in database and print out multiple copies. It works in both 64 and 128 (40-Column) mode. Anyone can surely find practical uses for Form Writer.

Special Delivery is a text and graphics C-64 adventure game that can provide hours of entertainment as you try to deliver a letter before it becomes too wet and melts. Since it is written entirely in Basic, you can easily examine the listing to learn a few tricks for writing adventure games of your own.

Now You See It is a joystick-controlled C-64 program from the June issue that challenges all your powers of concentration as you plot your way through numerous mazes in search of treasure.

To prepare for the next time your C-64 freezes up, we're offering June's Easy Applications program, Unlockup. It works by resetting your computer when you simultaneously press the control and restore keys. After the computer is reset, you can un-new the program that caused the computer to lock up in the first place.

Finally, you'll find two bonus programs on this issue of ReRUN. Math Whiz, a 64 and 128 (40-Column mode) game, hones your children's math skills by offering a wide range of exercises in the four fundamental arithmetic operations. The second bonus program is Space-Time Machine, for the C-64, which lets you experience the changes in time and mass that happen when you travel at or near the speed of light. You'll be amazed at the various changes that occur.

That completes the overview of this ReRUN issue. Before long, summer will be here, and I'll be back with our July-August edition. Stay tuned for more exciting program action.

Tim Walsh

Technical Editor
RUN *magazine*

How To Load

Loading from Menu

To get started, C-64 users should type LOAD "MENU 64",8 and press the return key. When you get the Ready prompt, the menu is loaded and you should type RUN to see a list of the programs on your disk. C-128 users need only press the shift and run-stop keys. When all the programs are displayed on the screen, you can run the one you select by pressing a single key.

Loading from Keyboard

If you do not wish to use the menu program, follow these instructions.

C-64: To load a C-64 program written in Basic, type: LOAD "DISK FILENAME",8 and then press the return key. The drive will whir while the screen prints LOADING and then READY, with a flashing cursor beneath. Type RUN and press the return key. The program will then start running. To load a C-64 program written in machine language (ML), type: LOAD "DISK FILENAME",8,1

C-128: All C-64 programs can be run on the C-128 as long as your computer is in C-64 mode. All C-128 programs are clearly labeled on the directory page. Your C-128 *must* be in C-128 mode to run these programs. To load a C-128 mode program, press the F2 key, type the disk filename and then press the return key. When the program has loaded, type RUN.

Making Copies of ReRUN Disks

Many programs on your ReRUN disk have routines that require a separate disk onto which the program writes or saves subfiles. To use these programs, you must first make a copy of the original program onto another disk that has enough free space on it to hold these newly written subfiles.

It's simple to make a copy of a Basic program. Just load it into your computer as outlined above, and then save the program back onto a separate disk that has plenty of free space for extra files.

Copying an ML program is not so simple. You cannot simply load and save an ML program; you'll need to use a disk-backup utility program, such as the one on your Commodore Test Demo disk.

Directory

Page	Documentation	Disk Filename	File Type
		* MENU 128	BASIC
		MENU 64	BASIC
2	Moving Messages	MOVING MESSAGE 1	BASIC
		MESSAGE 2	BASIC
4	Drip-Drop	DRIP DROP	BASIC
5	Graphix to the Max	* GRAPHIX MAX DEMO	BASIC
		* SWITCH DATA	BASIC
		* SWITCH	ML
7	Tick-Tock 128	* TICK-TOCK 128	BASIC
8	Scratch and Save	SCRATCH & SAVE	BASIC
9	Form Writer	FORM WRITER 64	BASIC
		* FORM WRITER 128	BASIC
		* EDITOR128 MAKER	BASIC
		EDITOR64 MAKER	BASIC
		EDITOR.64 ML	ML
		* EDITOR.128 ML	ML
16	Special Delivery	SPECIAL DELIVERY	BASIC
16	Now You See It	NOW YOU SEE IT	BASIC
17	Unlockup	UNLOCKUP	BASIC
19	£ Math Whiz	MATH WHIZ	BASIC
21	£ Space-Time Machine	SPACE TIME MACH.	BASIC

* - C-128 mode only
£ - Bonus program

Before you run a program, carefully read the documentation that pertains to it.

RUN it right: **C-64**

Moving Messages

By Mike Richardson

HAVE YOU EVER WISHED you could scroll large messages across your C-64's monitor screen—perhaps at a user's group meeting or in a class at school? Well, you can do it now with my Computer Board 64 program.

Moving Messages is a Basic loader for the main machine language program that's stored in memory locations $C000–$C4FF (49152–50431). The text you want to scroll is stored in $C500–$CFFF (50432–53247).

The machine language program can be activated by either of two SYS commands. SYS 49276 is the normal choice, because it uses Commodore's standard character set, but you can also load a custom character set into location 8192, then enter SYS 49284.

You'll find that Message Demo makes easy work of entering your message text into memory. All you have to do is set A$ to a line of text and enter GOSUB 60000. If you need assistance, refer to the REM statements.

Message Demo also contains information on variables used within Moving Messages that can be changed. Message Demo explains what to do—just follow the prompts.

You'll quickly discover that you can change the size of your characters, delay the scrolling and otherwise manipulate your message by using the function keys. The actions they perform are listed in Table 1.

How It Works

If you enter SYS 49276, the machine language program begins by copying the standard Commodore character set, then setting the screen colors. To set the colors without copying the character set,

use SYS 49284. You can change the starting values for the screen colors with the following Pokes:

Border/background—POKE 49296,color
Characters—POKE 49247,color

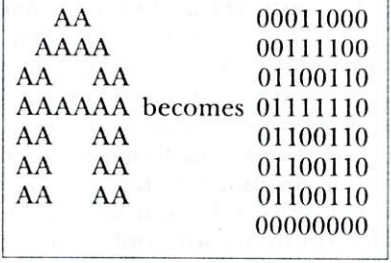
With these preliminaries out of the way, the program clears the screen and begins to process your text, starting with the first character. Characters on the C-64 consist of 8-pixel × 8-pixel matrices, so the program encodes each character as 64 ones and zeros (see Figure 1). It stores the code, then proceeds by columns and rows. If column 1, row 1 is zero, a space is printed to the screen; if it's a one, a character is printed. If the height is two, the last step is repeated once; if it's three, the last step is repeated twice. The program processes each row in turn, until row 8 is completed.

Next, the screen scrolls, and the last column is reopened. If the width is greater than one, the previous column is repeated until the

Table 1. Function Key actions.

F1	Change height
F2	Delay scroll
F3	Change width
F4	Change character
F5	Clear screen
F6	Repeat entire message
F7	Erase screen by scroll
F8	Exit to Basic

Figure 1. Sample character coding.



correct width is reached. The columns are processed in order until the character is finished; then it goes on to succeeding characters until your entire message is "on the move."

RUN it right: **C-64**

Drip-Drop

By John Fedor

WHATEVER SCENARIO, whether farfetched or realistic, you may supply to go with it, Drip-Drop is a fun game your whole family will enjoy. The action is uncomplicated, but the 25 difficulty levels offer plenty of arcade-type challenge.

Playing Catch

When you run Drip-Drop, the game screen immediately appears, with a status line and instructions at the top. Move the joystick in port 2 up and down to choose your starting difficulty level (1–25); then press the joystick button to start play at the level currently displayed.

The instructions disappear, and a dark cloud starts floating across the sky, just below the status line. Soon rain begins to fall from the cloud onto flames licking along the bottom of the screen. Three buckets are poised, one above the other, between the sky and the flames, waiting to go into action.

Use the joystick to move the buckets left and right and catch the falling drops. Each time you catch one, you're awarded five points. If you miss a drop, it hits the fire, the top bucket disappears and play continues with only two.

When you've missed three drops and lost all three buckets, the game is over, but you can start over by pressing the joystick button again. If you catch all the drops at one difficulty level, a short tune plays and you're sent to the next level.

The number of drops that fall at each level is equal to ten times the number of the level, for a maximum of 250. The higher the level the faster the drops fall and the greater the distance between them.

In the more advanced stages of the game, you'll wish you were toting around a swimming pool.

If you need a break during play, press the shift-lock key. The action will stop, and a blue border will appear around the screen. When you're ready, press shift-lock once more to continue. The best time to pause is between levels, just before it starts raining again.

I designed Drip-Drop to have as short a listing as possible while still providing interesting graphics, sound and action. The graphics consist of redefined graphics characters—six in the cloud, one in the raindrop, three in the bucket and four in the flames. The flame characters are continually redefined to simulate motion.

Oh-oh! I think I hear thunder. Better grab your buckets and get ready!

RUN it right: **C-128**

Graphix to the Max

By Lou Wallace

COMMODORE DESIGNED THE C-128's 80-Column mode for text only, but the 80-column chip (the 8563 video display chip, or VDC) has many properties pure text doesn't exploit. Switch responds to this unfulfilled potential by letting programmers use Basic 7's drawing commands to draw on the 40-column screen, and then switch that screen to 80-Column mode for display.

Challenges of Switching

I met a number of difficulties in writing Switch. First, since the 40-column graphics screen is 320 dots wide and the 80-column is 640 dots, an image switched as is would cover only half the screen and look distorted. To make the display look the same in 80 columns as in 40, I made Switch stretch the width of the picture from 320 to 640 dots.

Another problem concerned RAM. To make a 640 × 200 color display with 8 × 8-pixel color cells requires 18,000 bytes. C-128Ds have 64K of VDC RAM, but flat 128's have only 16K (unless you've upgraded to 64K). To provide some RAM for color information,

Switch creates a 640 × 176 display for 16K users. That means they can switch a 40-column screen only up to line 175, thereby losing 24 scan-lines of display space, but the benefit is worth the sacrifice. If you have 64K of VDC RAM, Switch creates a full 640 × 200 color screen. The program accepts a parameter (see below) that specifies how much RAM you have.

If you use the C-128's 80-column display, you may have guessed another problem I confronted in writing Switch: the fact that the 16 colors in the 40- and 80-Column modes are not exactly the same. Eighty-Column mode has two shades of cyan and two of purple, but no orange or dark gray. It also lacks brown, having a dark yellow instead. I mapped the colors as closely as possible, but, when switched, orange appears as dark purple, brown as dark yellow and dark gray as dark cyan.

There was no solution to the last challenge: that the 8563 VDC has no equivalent to Multicolor mode. With Switch, therefore, you may only use Graphic 1 (Hi-Res) mode.

Doodling

Besides graphics generated by your own programs, Switch will display Doodle! pictures. To use a Doodle! picture on the C-128, load it into memory location 7168 with a BLoad command, as in the following routine:

```
10 GRAPHIC 1,1
20 BLOAD "DDPICTURE",B0,P7168
30 GRAPHIC 5
40 SYS 2816,MD:REM SWITCH
50 GETKEY KE$
60 SYS 2819:REM 80-COLUMN GRAPHICS OFF
70 END
```

If you have hi-res pictures without color information, load them in directly at 8192, instead of 7168.

The Programs

Switch is a machine language binary file that other programs can load. Switch Demo uses several Basic 7 drawing commands in several colors. If you have a C-128D with 64K of VDC RAM, change the variable MD to 1; otherwise run Switch Demo as is.

Switch resides at address $0B00 (decimal 2816) and includes two main routines. SYS 2816,mode is the switch routine itself, with the

value of "mode" set at 0 or 1, depending on whether you have 16K or 64K of VDC RAM. SYS 2819 turns off Graphics mode and reinitializes the character sets and text screen attributes.

RUN it right: **C-128**

Tick-Tock 128

By Jesse Sherwood

C-128 CLOCK ADVANCES mankind's timekeeping ability by visually constructing two clocks on the C-128's 40-column screen.

One is a 12-hour analog clock with hour and minute hands and an AM/PM indicator. It's enclosed in a handsome "ebony" case, but if you'd prefer "mahogany," all you have to do is change line 310 to COLOR 1,10. This clock also strikes the hours and plays the authentic Westminster Chimes tune on the quarter hours.

Fitted into the base of the mantel clock is a 24-hour digital clock that displays hours, minutes and seconds.

Run Tick-Tock 128 in 80-Column mode. Set the time and start the clocks as prompted. When you switch your monitor to 40 columns, lo and behold, the clocks appear!

If you have 40-Column mode only, wait about 20 seconds until the program creates the initial drawing (nothing will be visible), then type in the current time using a 24-hour hhmmss format. For example, 2:30 PM would be entered as 143000. There still won't be any display, so you'll have to do this "in the blind." Finally, press return to see the clocks.

Since the display is updated every second or so, you may wonder why it appears to be continuous. Here's how I achieved that effect. After selecting Bit-Mapped mode, the video memory is moved to another video bank, and a screen of data is created in the bit-mapped location. Next, a machine language subroutine switches that block of data to the new location—too quickly to be noticed. At this time, the old screen is erased and a new one starts replacing it in the same location. The result is a display that doesn't wink, blink or otherwise appear corrupted. However, rounding off pixels occa-

sionally gives small odd movements to the hands.

In early versions of C-128 Clock, I derived the time from Basic's TI$ function. However, TI$ isn't very accurate, and I noticed that the clocks were losing time, so I switched over to the 6526 CIA-1 chip, which derives the time from the ac powerline frequency.

Foreign users please note that bit 7 in location 56334 must be set for 50 Hz. In line 420, replace PEEK (C+6) AND 127 with PEEK (C+6) OR 128.

Now, who will be first to add a calendar to the C-128 Clock? Or an alarm? Or a control circuit for the coffee pot? Or perhaps some hams or aviators might like the digital readout to display Greenwich Mean Time?

RUN it right: **C-64; 1541 disk drive**

Scratch and Save

By David Archibald

DO YOU HESITATE to use the 1541's Save with Replace command because it might corrupt your disk files? If so, you're well aware of how risky the Save@ command is. But playing it safe is a problem, too, because the only other way to update a file and retain the original filename is to go through the work of deleting the old file first. Computers are supposed to relieve us of tasks, not give us more!

I wrote Scratch and Save to provide relief from the delete-file/save-file routine. Before it saves with a filename that already exists on the disk, it automatically scratches the old file for you. In essence, it performs the same function as Save with Replace, but avoids the danger of ruining your disk.

No special commands are needed to use Scratch and Save. In fact, the syntax is exactly the same as the normal SAVE "FILENAME",8.

Scratch and Save is compatible with fast-load cartridges, such as Epyx's Fast Load and Access Software's Mach 5. You should also be able to use it with any other program that saves files by using the Basic Kernal's Save command, as most programs do.

You can disable Scratch and Save by simultaneously pressing the run-stop and restore keys, and then reenable it by entering SYS 828.

RUN it right: **C-64; C-128 (40-Column mode)**

Form Writer

By Tom Brown

IF YOUR FAMILY, small business or club runs on a limited budget, you can save money by using Form Writer on the C-64 or 128 (in 40-Column mode) to create single-page forms for many purposes—letters, invoices, questionnaires, membership records, and so on. The program allows you to print custom-designed forms, containing fields that you can fill in by using the keyboard or the program's document-oriented database.

Each form contains 50 lines, each with 80 characters, and you can prefix any line with one of nine print codes, allowing condensed, italic or double-sized type. Because these codes are defined in the Basic portion of the program, you can easily customize them to suit your needs. There's even provision to send up to ten nonprintable characters to the printer, so you can send special printer codes within lines.

First Steps

Be sure to run the C-128 version in 40-Column mode. When the program is activated, it first prompts you to enter the month and day, to be used when there's a date field in your document. Next, the program checks the disk drive, so all future disk operations default to that drive. You can easily change the drive by using the Disk Drive option on the main menu.

Creating and Editing Blank Forms

After the preliminaries, you're ready to create a blank form. On this form, you'll place the text, which won't change from document to document, and the form fields, which will.

At the main menu, select option 1, the Form Editor. It will take you to the Form/Field menu, where you should select option 1 again, to create a new form. This selection erases any old form in memory. If, instead of creating a new form, you want to edit the one currently in memory, choose option 2 at the Form/Field menu.

The editor functions within a 40-column window on an 80-column

by 50-line page. Using the cursor keys, you can move this window to cover any area on the page.

Whenever you enter the editor, you'll find the blinking cursor in the first column on the first line. At the bottom of the screen there's a status line that shows the position of the cursor in the 80-column form (not in the 40-column window). Both lines and columns are numbered starting at zero. As you type characters or use the cursor keys, the status line updates automatically. You'll find this handy for keeping tabs on the cursor's location when it's traveling a distance on the page and seems to vanish. Forms can scroll horizontally in the window as far as the 80th column (column number 79), which represents the edge of the paper.

As you're working, press the return key to advance the cursor to the beginning of the next line. To move the cursor right several spaces, press the tab key on the C-128 or control/I on the C-64. The insert and delete keys operate in the normal fashion, but affect the entire 80-column line, not just the 40 columns in the window. *Don't* use the insert key when the cursor is in the first column of any line. Also, the home key sends the cursor to the top of the form, not the top of the window.

To erase the screen on the C-128, press the alt key along with clear-screen (shift/home). On the C-64, clear-screen doesn't work at all; you must go back to the Form/Field menu and select option 1 (create a new form).

To exit the editor, use the escape key on the C-128 or the F1 key on the C-64.

Creating Fields

To create a field anywhere on your form, define its start and end positions by pressing the up-arrow key where the first and last characters will be. Unpaired up-arrows are ignored. The program allows a maximum of 256 fields on a page, but memory limitations will undoubtedly lower that number, particularly with the C-64.

Use the left-arrow character to direct Form Writer to follow one of the special printer code sequences in line 6 of the listing. The program defaults to printing "hello" when it encounters this character, but, if you change line 6, it will print whatever you wish, within limits. The printer code cannot include more than ten characters, nor can it contain any zeroes. A CHR$(0) is required at the end. The CHR$(0) is not sent to the printer, but is used by Form Writer to locate the end of the printer code.

Only one code sequence can be used at a time in a document, but it can appear as many times as you wish. Don't confuse these codes with the commands for changing printer fonts that go at the beginning of lines (see Printing, below).

When you exit the editor, it asks if you want to define the fields you've just created. If you're not satisfied with them yet, select option 2 in the Form/Field menu to edit the form currently in memory. Skip that option's erase-form step and go directly to the editor to make your changes. When you exit option 2, you can define the fields you passed over before.

If you opt to leave the editor without creating any fields, you go directly back to the Form/Field menu.

Defining Fields

Once you have entered your fields into the form, the program proceeds through them from top to bottom and left to right on the page, displaying the number of each field (starting at zero), its line and column position and its length, and you must give each field a name. For those fields you'll be filling yourself, as opposed to those the computer will fill, this name should be meaningful, such as Last Name or Address.

After you've named the field, you must specify the type of data it will hold, using the following list of ten possibilities:

Alphanumeric: Any printable character, including numbers, but the field cannot be used for calculations.

Sum1, Sum2 and **Sum3:** Numeric values, which will be added into three subtotals.

Total1, Total2 and **Total3:** Numeric values—these fields hold the subtotals of the three Sum fields listed above.

Grand Total: Numeric value—the total of the Total fields above. If you wish to subtract any Total field from the others in reaching the Grand Total, enter negative values in the appropriate Sum fields.

Keyboard: Data typed in by the user at the time of printing.

Date: The date you type in when you run Form Writer.

Data to fill Alphanumeric, Sum and Keyboard fields is entered by the user. Data for all the other types of fields is entered automatically by the computer when you print your document.

Be aware that you can't use the editor to preview a filled-out document before printing. Prior to printing, the blank form is copied to another area of memory (with the original left intact) where its fields are filled. The editor cannot view that area.

Should you want to abort defining fields—if, for example, you gave the last field the wrong name or type—press return at any field-name prompt to exit to the Form/Field menu.

Creating the Database

After you've defined all your fields, Form Writer presents the name of each one and asks in what order you want the program to prompt you to fill them in, because it's seldom convenient to do so in the same order the program numbers them. For example, the form may have a person's last name listed before his or her first name, but the reverse order would be easier to key in. There's no error-checking here. If you enter the same priority for two or more fields, only the lowest-numbered field with that priority will be used; the database will skip over the others. Be careful to designate each priority only once, and you'll have no difficulty.

When this step is done, enter a name for your database. The name can be no longer than 15 characters, because the program prefixes it with a slash mark. The slash mark distinguishes database files from document files. Don't enter this slash mark yourself when reading the database back; the program will do it for you.

Saving and Loading the Form

Don't forget to save your form after you've edited it! Also, note that you can use a form with several different databases, but you must be sure to have the correct form loaded into memory before defining fields or printing a document. Saving and loading are done from the Form/Field menu.

Using the Database

Once you've designed your blank form, defined its fields and saved your database to disk, you're ready to use the Database option in the Main menu to view, add, change or delete the user-entered Alphanumeric or Sum fields. If you don't have a database open at this point, the program forces you to open one in order to reach the Database menu. From that menu, you can easily change databases with the Open Database option. The name of the currently open database is listed on the Database menu screen.

To Form Writer, the database is an outline, or template, of the length and type of each field in your blank form. Each record in the database is referred to as a document. Rather than use relative files, which would be slow and awkward in this application, each

document is stored in a sequential file. These files are very small and can be easily copied (using appropriate software) to backup disks or transmitted via modem. They also give you the advantage of giving a name to each document, rather than dealing with it only as a record number.

Deleting a document involves nothing more than scratching its sequential file. The program pauses and tells you to hit "any key to continue," after listing the Files Scratched message. The Edit Document option displays the same message, since it deletes the old document before saving the new one.

Disk Drive Help

It's virtually impossible to remember the filenames for a whole collection of databases, documents and forms, so you need to have access to your disk directory as you're using Form Writer. When you're in the middle of working on something, you can view the directory by entering a dollar sign as the filename at many filename prompts. Then, after you're done, the program returns you to the same prompt.

At those times when you're free to go to the Main menu, you can view the directory through the Disk Drive option, which provides access to the disk drive error channel, as well. Drive commands are standard DOS Wedge commands, only you can omit the @ at the beginning. Entering a number higher than 7 instead of a disk command changes the default drive setting. Since the program defaults to the drive it was loaded from, you can use this option to change to another drive (or even the RAM DOS!) for your data storage.

Printing

When it comes time to print out your form, choose the Printer option in the Main menu. If you haven't opened a database and loaded a form at this point, the program will ask you to do so. There are also menu options to change the database and form.

You can print a form using a document from your database, or you can enter the information from the keyboard (but only information that would normally be stored in the database). There's another option for printing out a form with the fields left blank, so you can fill them in by hand. In this case, the fields are printed with dotted lines. Remember to leave enough vertical spacing in the blank form to give your penmanship room to breathe.

Before you print out your form, there's one last thing you might want to do: Call on some of your printer's special effects to create a more professional look. You can use Commodore graphics characters (if your printer or interface permits it), but perhaps you'd like to use italic, double-width or condensed characters. Select the Set Printer option from the Printer menu, then enter your printer codes. This is also where you can change the printer device numbers and secondary addresses.

Note that two printer files are opened for printing, one of which has a default secondary address of 7 for printing lowercase letters. This is the channel that each line of the form is sent through. The other channel is opened to secondary address 5, which, on most printer interfaces (such as the Cardco and Xetec), lets you send printer command codes without fear that the interface will think they're regular characters and translate them. You can change one or both secondary addresses, depending on your system and your needs.

Next, you get a chance to define a printer code to be sent at the start of each line. The default codes are for the Gemini 10X. All you have to do is change the code, plus the prompt on the line beside it, to match your system's capabilities. This code applies only to the line in question.

As you type in Form Writer, have your printer manual handy, because some of the Gemini 10X commands may not be available on your printer. On the Gemini 2, for example, the italic character set has been replaced with a near-letter-quality (NLQ) font. In most cases, however, all you need to do is change the codes that perform each task. If you own an Epson MX-80-compatible printer, you may find a minimum number of changes necessary. If yours is an exotic printer, you may have to perform major surgery to achieve special effects.

The printer commands are sent to the printer at the start of each line along the transparent secondary printer channel. Printer commands you send using the left-arrow character embedded in your form, on the other hand, are sent along the text channel, so they may be translated by your interface into true ASCII. Keep that in mind when changing the Form Writer listing.

One warning about embedded code: If you send a string of printable characters, the line printed will be longer than 80 characters. This is of no concern when you're printing in Condensed mode, sending *non*printing control characters to activate a special

effect or sending a code that prints a single printing character (such as a user-defined character).

As I mentioned earlier, the embedded codes are located in line 6 of the program listing. The ten codes you can elect to send at the start of a line are located in lines 2820–2830.

The program always prints 80 characters to a line, so, if you plan to use an enlarged character size, remember to make the line shorter! There's no way to make a line longer than 80 characters to take advantage of smaller character sizes. The default printer codes reset the printer at the start of each line, so any special formatting must be done for each line individually.

When you're done defining your specially formatted lines, press return at the Select Line prompt to go back to the Set Printer menu. At this point, I'd suggest that you *save* the printer codes you've defined, because printer codes are erased whenever you open a database.

You can use a pattern of printer codes with more than one database, just as you can use a database with more than one blank form (and vice versa). The Set Printer menu also lets you erase all printer codes in memory (as is done when you choose to set the codes) and load a set of codes you've previously defined. Saving printer codes also saves the printer device number and secondary address. Press return to go back to the Printer menu.

When you finally get around to printing your form, you can specify the number of lines to print, which is handy if the form is shorter than a full page. You can also print several copies without going back to the Printer menu.

Parts of the printing process are slow, particularly if you have a lot of fields to be generated. This is partially due to the fact that the field contents must be padded with spaces before being sent to the form. A little patience here goes a long way.

Technical Notes

The C-64 version of Form Writer uses the area under the Basic ROMs to hold the blank form, while the C-128 version uses the hi-res screen area. Lines 65–67 of the listing may seem strange: They undimension all arrays in memory (without losing any other variables!) whenever you open a database. In this way, smaller databases can have smaller arrays, and so take up less memory. Saving memory is unimportant on the C-128, but on the C-64 it eliminates a lot of needless delay due to garbage collection.

RUN it right: **C-64**

Special Delivery

By Cindy Hurley

IN SPECIAL DELIVERY, you are a dedicated postman, charged with carrying a special delivery letter to the crazy hermit who lives in the woods. To complete your assignment, you must evade a threatening snake, a booby trap, a persistent dog and a mettlesome rabbit. You must also take care to prevent the rain from soaking the letter. The final obstacle is the crazy old man himself, who might shoot you if he's in a bad mood.

The game can understand two-word inputs and has a dictionary of over 30 words. You may visit 20 different locations, and take and drop various objects, including the special delivery letter. The game responds to whatever you type with an appropriate comment, ranging from "I don't understand" to "I see no mailbox here."

This is a nonviolent adventure, in that you can't kill anything. On the other hand, *you* can be killed if you're careless. Therefore, you must use all your wits to overcome the obstacles; tools to help are scattered throughout the woods. If you fail, just start the game over and try again. Good luck!

RUN it right: **C-64; joystick**

Now You See It. . .

By Penny DeGroff

YOU PAY $30 for a popular arcade game, and, in spending several hours or days playing it, you just can't get past the third level. If

this sounds familiar, you'll enjoy playing Now You See It. This game has only eight levels, and Carkey, the main character, is indestructible. All he has to do is walk around inside Tinshor Cave and pick up pieces of yendor, a valuable metallic substance. There are ten pieces of yendor on each of the eight levels. Sounds easy, right?

Wrong! If you've ever been in a cave, you know that the farther you go, the darker it gets. The first level of Now You See It is a cinch. All Carkey has to do is to tour the cave, guided by a joystick in port 2, and pick up the pieces of yendor. As he proceeds, the number in the upper-right corner of the screen displays how many pieces remain.

After collecting all ten pieces, Carkey automatically descends to level 2. Since it's darker there, it's harder to see the passages. If Carkey keeps bumping into walls, press the fire-button. This lights the cave, but it also freezes Carkey in his tracks. Release the fire-button, and the walls disappear again.

All the time that Carkey is moving through the cave, the numbers in the upper-left corner of the screen tell how much time has elapsed. Once the clock starts, it doesn't stop until all eight levels have been completed.

In each of the lower levels, different things or combinations of things are obliterated by the darkness. If this still sounds like an easy game, try to play it in five minutes or less!

RUN it right: **C-64**

Unlockup

By James Host

ONE OF THE MOST TRYING experiences a Commodore 64 user faces is lockup, which, as the word implies, locks up, or freezes, the

program in memory. The program refuses every effort on your part to resume running, the run-stop/restore combination has no effect, and the computer doesn't respond to any keypresses. Sometimes a lockup displays the Ready prompt with a flashing cursor, but the computer is unresponsive to any Basic Direct mode commands.

In any case you're forced to perform one of two actions, and both are unpalatable: reset the computer or turn it off and on again. In either case, you lose the Basic program in memory.

Unlockup gives you both a soft reset and an un-New feature. With Unlockup, you can *usually* recover a Basic program.

Unlocking Unlockup

Although the active portion of Unlockup is written in machine language, the ML is encoded in Data statements.

When you run Unlockup, it's installed at the top of Basic memory and executes in the background while other programs are running. Load and run other programs as you normally would.

Now you're prepared for any lockup! When it happens, hold down the control key and tap restore. The computer will reset, and the C-64 power-up message will appear (with fewer bytes free, of course), along with the prompt, Do you want to un-New (y/n)?

To recover the Basic program, type Y; you'll see the Load command, but nothing will be loaded. Instead, Basic is working to restore the lost line-links in the program you're recovering. Following the un-New operation, you can list and save your Basic program normally. In fact, if at some time you accidentally type NEW, you can recover your program by hitting control/restore and answering Y to the un-New query.

Other Useful Features

In addition to its crash-recovery features, Unlockup re-enables the stop key and run-stop/restore combination (but is not itself disabled by run-stop/restore) for Basic programs that would otherwise be disabled. You can also set Unlockup's default colors for Basic's Direct mode if you're tired of the C-64's default blue characters, screen and border colors—just change the numbers in the REM Data statements near the end of the Unlockup Basic loader. The current numbers display a dark-gray text, medium-gray border and light-gray background.

RUN it right: **C-64; C-128**

Math Whiz

By Dan S. Campbell

MATH WHIZ HELPS SHARPEN your basic math skills. The program offers addition, subtraction, multiplication or division, plus a combination of addition and subtraction or of all four functions.

The numbers are oversized and green, making for easy and fast identification. As attention-grabbers, the program features color and sound. For some added excitement, you race against a clock located at the top of the screen.

In my home, children of ages five to ten have played the game with equal interest; but adults enjoy it, too. Racing the clock for a higher score keeps everyone coming back. (Don't be surprised if it soon becomes difficult to keep up with the kids).

There are 15 questions per round, and you have up to 15 seconds to answer each question. Always press the return key after entering your answer. Speed is rewarded by a higher point total, but all correct answers are rewarded with music and a flashing screen.

How the Program Works

To display oversized numbers, I printed string variables, then placed them in an array. I also stored the function symbols in an array.

I used the Commodore's graphics characters along with the cursor keys to design numbers 0–9. I placed each figure in a string array with corresponding subscript numbers. The routine on line 1100 uses the initial ten figures to piece together the remaining numbers up to 81 and to place them in the array. I also drew a blank figure, which acts as an eraser if you press the delete key.

The variables A and B are single-digit numbers chosen at random, and they form the basis for each problem, no matter which function is chosen. X is the sum of A and B, and Y is their product.

Addition or multiplication problems are set up as A + B = X or

A × B = Y, and you must find X or Y, depending on the problem. Subtraction or division problems are set up as X − B = A or Y ÷ B = A, and you must solve for A in each case.

A string variable is assigned to each numeric variable, and the problem is then printed on the screen by the subroutine at line 400 or 500.

I used a Get loop, rather than an Input statement, to retrieve answers; in this way, the program can continually pass through the timing routine. The first loop checks for a number or a return. If a number was entered, the appropriate string is retrieved and printed.

The next loop then checks to see whether the delete key was pressed. If it was, the original problem is reprinted with the answer erased. If the return key was pressed, the answer is compared with the correct answer in the subroutine at line 700.

If you answer incorrectly, you'll hear an unpleasant buzzer, and the answer will be displayed. If you answer correctly, you'll be treated to a tune and a flashing screen.

At the beginning of each problem, the computer's internal clock, TI$, is set to 000000. The subroutine at line 600 then subtracts the value of TI$ from 15 and prints the amount of time remaining at the top of the screen. If the time remaining equals zero, then the program moves to the wrong-answer routine at line 800.

The variable TI is updated 60 times a second by Commodore's internal clock. In the event of a correct answer, line 910 uses this feature to augment your score by one point for each thirtieth of a second remaining on your time. On the other hand, a wrong answer will reduce your score by 250 points.

If you wish to change the time allowed for each answer, change the value 15 in line 600. Remember, however, to change the value in line 910 to correspond with your new time. (The value of 900 was determined by multiplying 60 times the desired time allowed for each problem.) You may change the number of problems in a round by changing the value of R in line 100.

The program plays music by reading and poking a note at a time from the data table. Following each note is a value for its duration; this is placed into a timing loop. After each correct answer, the data is restored to play the same part again. If your answer to the last problem is correct, the song is continued to the end. If it is incorrect, the flag (variable F) causes the data to be restored in line 1210, and the song is then played from the beginning.

RUN it right: **C-64**

Space-Time Machine

By Dieter Bohr

WHAT REALLY HAPPENS WHEN a spaceship approaches the speed of light? Some of the books dealing with Einstein's Special Theory of Relativity explain the consequences of high-speed motion with complex mathematical formulas and diagrams, in an attempt to arouse feelings of wonder in the reader. Yet, no matter how clear and provocative these math functions are, they lack the presentations needed to make exciting the phenomena they model.

To create that excitement, I've written Time-Stop Spaceship, a program that demonstrates the results of these equations by simulating the dynamics expressed in them. The program lets you explore time and distance with a journey into the future without adding to your age, lets objects shrink while their masses increase manyfold and slows down your time while everyone else's marches on.

The program uses the "two-clock" scenario: You are the pilot of a spaceship whose clock is synchronized with an identical clock on Earth at Mission Control. You can accelerate your spaceship to almost the speed of light, remain at this velocity for a period of time, then turn around. You accelerate again to light-speed velocity for the return trip. When you're travelling, you can simultaneously view both the earth clock/calendar and your own clock/calendar, and, lo and behold, they're no longer synchronized.

You may only be gone for a few months. However, depending on how fast you've travelled, you may find you've returned to Earth hundreds of years in the future. In fact, you might land in a cornfield because Mission Control no longer exists.

Simulation Time Scale

The simulation time-scale factor is a value between 1 and 10800. It can be used to speed up the simulated events from normal (factor = 1) to high speed (factor = 10800). The time-scale factor of 10800 lets you simulate a three-months-long period in a span of about 12 minutes. When a factor of 1 is entered, one second on the spaceship clock corresponds to one second on your wristwatch. Any

other value will multiply the time intervals on the spaceship in proportion.

For example, a factor of 60 speeds up events 60 times, so that one second on your wristwatch (real-time) corresponds to about one minute on the spaceship's clock; a factor of 10800 makes one real-time second equal to about three hours on the spaceship's clock.

All the simulation parameters automatically default to preset values if you just press the return key after the prompts. Once all the simulation parameters are entered, the main simulation screen appears, showing the spaceship at its initial state on Earth, with its engines idling and its clock synchronized with that of Mission Control.

You start the ship moving by pressing the plus (+) key. Each stroke of this key increases the rate of acceleration by one "g." G represents the rate of acceleration due to the force of gravity on Earth. Its value is approximately 9.8 meters (.0098 kilometers, or 32 feet) per second per second. This is the acceleration experienced by a body that is falling freely near sea level, without considering the effects of air resistance.

In a symmetric fashion, pressing the minus (−) key reduces the acceleration by one g. This produces a braking (decelerating) action and is used to reduce the forward velocity of the ship.

Pressing the up-arrow key either doubles the acceleration rate or halves the rate of deceleration, depending on whether you were last increasing with the + key or decreasing with the − key. The acceleration indicator can be increased to a maximum of 3,200,000 with a simulation factor of 1, or to 296 with a simulation factor of 10800. These limits avoid program variable overflows.

Since the numbers on the screen change rapidly, you can freeze the action by pressing the space bar. Once the screen is frozen, pressing the space bar or any other key causes the program to resume. Pressing the space bar twice in quick succession advances the simulation by one set of computed values.

The return key turns the spaceship around for its trip back to Earth. When you press this key, the screen border turns light red to indicate that you're ready to start on the return trip. The clocks operate at their normal rate again, and the spaceship is resting and poised for the second blast-off. Return can be pressed at any time, but for more realism, you should slow the ship's movement with the − key (just as you'd brake a car before turning around), and then press return.

I hope you enjoy your journey into space and time. ■

11 Programs Included on this Disk

Moving Messages ► C-128 80-Column Hi-res Converter
C-128 Clock ► Bug-Free Save-With-Replace ► Form Writer
Unlockup 64 ► Special Delivery Adventure Game

From the May RUN:

- Moving Messages
- Drip-Drop
- Graphix to the Max
- Tick-Tock 128
- Scratch and Save

From the June RUN:

- Form Writer
- Special Delivery
- Now You See It
- Unlockup

Plus: Extra Bonus Programs!

- Math Whiz
- Space-Time Machine

If any manufacturing defect becomes apparent, the defective disk will be replaced free of charge if returned by prepaid mail within 30 days of purchase. Send it, with a letter specifying the defect, to:

ReRUN • 80 Elm Street • Peterborough, NH 03458

Replacements will not be made if the disk has been altered, repaired or misused through negligence, or if it shows signs of excessive wear or is damaged by equipment.

The programs in ReRUN are taken directly from listings prepared to accompany articles in *RUN* magazine. They will not run under all system configurations. Use the RUN It Right information included with each article as your guide.

TECHNICAL EDITOR
TIM WALSH

MANAGING EDITOR
SWAIN PRATT

COPY EDITOR
PEG LEPAGE

PROOFREADER
HAROLD BJORNSEN

DESIGN AND LAYOUT
ANNE DILLON

TYPESETTING
DEBRA DAVIES
BETH KROMMES
KEN SUTCLIFFE

RERUN MANAGER
PAUL FINCH

CUSTOMER SERVICE REPRESENTATIVE
DEBBIE BOURGAULT